Claims

Sub A1

[c1] 1. A method for cache management comprising:
requesting database lock of a named cache;
locking the named cache and providing an indication that the named cache is locked;
acquiring a local lock of the named cache at a local node;
generating a timestamp corresponding to the local lock;
invalidating a cache item of the named cache in the local node;
releasing the local lock of the named cache;
sending a message to a remote node identifying the cache item of the named cache;
receiving acknowledgment from the remote node;
sending an update of the cache item of the named cache;
updating the named cache; and
releasing the database lock of the named cache.

[c2] 2. A method according to claim 1, further comprising:
acquiring a local lock of the named cache at the remote node.

[c3] 3. A method according to claim 1, further comprising:
invalidating the cache item of the named cache at the remote node.

[c4] 4. A method according to claim 1, further comprising:
acknowledging the message identifying the cache item of the named cache.

[c5] 5. A method according to claim 1, further comprising:
releasing the local lock of the named cache at the remote node.

[c6] 6. A method according to claim 1, wherein locking the named cache locks the named cache for all nodes.

[c7] 7. A method for cache management comprising:
identifying a cache miss of a cache item;
requesting a read lock of a named cache, the named cache including the cache item;
read locking the named cache;

requesting the cache item from a master locking database;
receiving the cache item; and
releasing the read lock of the named cache.

[c8] 8. A method according to claim 7, further comprising sending an indication that the named cache is read locked.

[c9] 9. A method according to claim 8, further comprising sending the cache item from the master locking database.

[c10] 10. A method for cache management comprising:
identifying a cache miss of a cache item;
requesting a read lock of a global database, the global database including the cache item;
read locking the global database;
requesting the cache item from a master locking database;
receiving the cache item; and
releasing the read lock of the global database.

[c11] 11. A method for cache management comprising:
determining that a predetermined event has occurred;
requesting a read lock of a named cache;
requesting a timestamp;
receiving an indication of a read lock of the named cache;
receiving a timestamp;
comparing the received timestamp with a previous timestamp;
responsive to the comparison, performing a predetermined action; and
releasing the read lock of the named cache.

[c12] 12. A method according to claim 11, wherein the predetermined action comprises storing the received timestamp.

[c13] 13. A method according to claim 11, wherein the predetermined action comprises requesting an update of the named cache.

[c14] 14. A method according to claim 11, wherein the predetermined action

comprises receiving an update of the named cache.

[c15] 15. A method for cache management comprising:

determining at a local node that an update is required for a cache item in a named cache;

sending a request for a database lock of the named cache from a local node to a cache manager;

receiving the request for a database lock at the cache manager;

locking the database of the named cache;

sending an indication that the database of the named cache is locked from the cache manager to the local node;

receiving the indication that the database of the named cache is locked at the local node;

acquiring at the local node a local write lock of the named cache;

sending a timestamp from the local node to the cache manager;

receiving the timestamp at the cache manager;

updating a lock table with the timestamp;

invalidating at the local node the cache item of the named cache;

releasing the local write lock of the named cache;

sending a message to invalidate the cache item of the named cache from the local node to a second node;

receiving the message to invalidate at the second node

acquiring at the second node a remote write lock of the named cache;

invalidating at the second node the cache item of the named cache;

sending an acknowledgment of the message to invalidate from the second node to the local node;

releasing the remote write lock of the named cache at the second node;

receiving the acknowledgment of the message to invalidate at the local node;

determining that no further acknowledgments of the message to invalidate are expected at the local node;

sending an update of the cache item of the named cache from the local node;

receiving the update of the cache item and updating the cache item; and

releasing the database lock of the named cache.

[c16] 16. A method for cache management comprising:

identifying a cache miss of a cache item at a local node;

sending a request for a read lock of a named cache from the local node to a cache manager;

receiving the request for a read lock of the named cache at the cache manager;

read locking the named cache at the cache manager;

sending an indication that the named cache is read locked from the cache manager to the local node;

receiving the indication that the named cache is read locked at the local node;

requesting the cache item from a master locking database;

receiving the request for the cache item;

sending the cache item to the local node;

receiving the cache item at the local node;

sending a release of the read lock of the named cache from the local node to the cache manager;

receiving the release of the read lock of the named cache at the cache manager; and

releasing the read lock of the named cache.

[c17] 17. Computer executable software code transmitted as an information signal, the code for cache management, the code comprising:

code to request a database lock of a named cache;

code to lock the named cache and provide an indication that the named cache is locked;

code to acquire a local lock of the named cache at a local node;

code to generate a timestamp corresponding to the local lock;

code to invalidate a cache item of the named cache in the local node;

code to release the local lock of the named cache;

code to send a message to a remote node identifying the cache item of the named cache;

code to receive acknowledgment from the remote node;

code to send an update of the cache item of the named cache;

code to update the named cache; and

code to release the database lock of the named cache.

[c18] 18. A computer readable medium having computer executable program code stored thereon, the code for cache management, the code comprising:

code to request a database lock of a named cache;

code to lock the named cache and provide an indication that the named cache is locked;

code to acquire a local lock of the named cache at a local node;

code to generate a timestamp corresponding to the local lock;

code to invalidate a cache item of the named cache in the local node;

code to release the local lock of the named cache;

code to send a message to a remote node identifying the cache item of the named cache;

code to receive acknowledgment from the remote node;

code to send an update of the cache item of the named cache;

code to update the named cache; and

code to release the database lock of the named cache.

[c19] 19. A programmed computer for cache management, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in memory, wherein the program code comprises:

code to request a database lock of a named cache;

code to lock the named cache and provide an indication that the named cache is locked;

code to acquire a local lock of the named cache at a local node;

code to generate a timestamp corresponding to the local lock;

code to invalidate a cache item of the named cache in the local node;

code to release the local lock of the named cache;

code to send a message to a remote node identifying the cache item of the named cache;

code to receive acknowledgment from the remote node;

code to send an update of the cache item of the named cache;

code to update the named cache; and

code to release the database lock of the named cache.